rebmix: The Rebmix Package

Marko Nagode

September 5, 2014

Abstract

The **rebmix** package for R provides functions for random univariate and multivariate finite mixture generation, number of components, component weights and component parameter estimation, bootstrapping and plotting of the finite mixtures. It relies on the REBMIX algorithm that requires preprocessing, information criterion and conditionally independent normal, lognormal, Weibull, gamma, binomial, Poisson or Dirac component densities. The algorithm optimizes the component parameters, mixing weights and number of components successively based on the boundary conditions, such as the maximum number of components, total of positive relative deviations, number of classes or nearest neighbours. The algorithm is robust, time efficient and can be used either to assess the initial set of the unknown parameters and number of components for, e.g., the EM algorithm or as a standalone algorithm that is a good compromise between the nonparametric and parametric methods to the finite mixture estimation.

1 Introduction

Finite mixture models are used increasingly to model the distributions of a wide variety of random phenomena. For the multivariate data of continuous nature, attention is paid to the use of multivariate normal components because of their computational convenience (McLachlan and Peel, 1999; Ingrassiaa and Roccib, 2007; Frühwirth-Schnatter, 2006). However, in fatigue and reliability analyses, lognormal and Weibull distributions are preferred due to their flexibility and their definition for continuous positive random variables only (Majeske, 2003; Sultan et al., 2007; Touw, 2009).

The finite mixture models have seen a real boost in popularity over the last two decades due to the tremendous increase in available computing power. These models can be applied to data where observations originate from various groups and the group affiliations are not known, and on the other hand to provide approximations for multimodal distributions Leisch (2004). Some of the latest models can be found also in McLachlan and Lee (2013); Chavent et al. (2012); Grün et al. (2012); Melnykov et al. (2012); Ardia et al. (2009); Benaglia et al. (2009); Grün and Leisch (2008); Fraley and Raftery (2007); McLachlan and Peel (2000).

The REBMIX algorithm origins in Nagode and Fajdiga (1998) and avoids the drawbacks of the EM algorithm:

- The EM algorithm converges to a local maximum of the likelihood function very quickly.
- There are often several other promising local optimal solutions in the vicinity of the solutions obtained from methods that provide good initial guesses of the solution.
- Model selection criterion usually assumes that the global optimal solution of the log-likelihood function can be obtained. However, achieving this is computationally intractable.
- Some regions in the search space do not contain any promising solutions. The promising and non-promising regions co-exist, and it often becomes challenging to avoid wasting computational resources to search in non-promising regions.

reported in Reddya and Rajaratnam (2010) by updating the number of components, component weights and component parameters sequentially and not simultaneously (see also Celeux et al., 2001). Later on the REBMIX has evolved (Nagode and Fajdiga, 2000; Nagode et al., 2001; Nagode and

Fajdiga, 2006, 2011a,b). The paper extends it to discrete variables by adding binomial, Poisson and Dirac parametric families. Gamma parametric family is added as well. REBMIX stands for a robust, time efficient tool that can be used either to assess the initial set of unknown parameters and the number of components for other algorithms (Chen et al., 2013; Bučar et al., 2004) or as a standalone procedure that is a good compromise between the nonparametric and parametric methods to the finite mixture estimation.

The **rebmix** implementation of the REBMIX (Nagode, 2014) extends the set of algorithms available for random univariate and multivariate finite mixture generation, number of components, component weights and component parameter estimation, bootstrapping and plotting of the finite mixtures in the R language and environment for statistical computing.

The outline of the paper is as follows: Section Algorithm presents the algorithm. Section Examples analyses the performance of the approach by studying four datasets. Section Summary lists the conclusions.

2 Algorithm

Let $\boldsymbol{y}_1, \dots, \boldsymbol{y}_n$ be an observed d dimensional dataset of size n of continuous or discrete vector observations $\boldsymbol{y}_j$. Each observation is assumed to follow predictive mixture density

$$f(\boldsymbol{y}|c, \boldsymbol{w}, \boldsymbol{\Theta}) = \sum_{l=1}^{c} w_l f(\boldsymbol{y}|\boldsymbol{\theta}_l) \tag{1}$$

with conditionally independent component densities

$$f(\boldsymbol{y}|\boldsymbol{\theta}_l) = \prod_{i=1}^{d} f(y_i|\boldsymbol{\theta}_{il}) \tag{2}$$

indexed by vector parameter $\boldsymbol{\theta}_l = (\boldsymbol{\theta}_{1l}, \dots, \boldsymbol{\theta}_{dl})^\top$. The components can currently belong to either normal

$$f(y_i|\boldsymbol{\theta}_{il}) = \frac{1}{\sqrt{2\pi}\sigma_{il}} \exp\left\{-\frac{1}{2}\frac{(y_i - \mu_{il})^2}{\sigma_{il}^2}\right\},$$

lognornal

$$f(y_i|\boldsymbol{\theta}_{il}) = \frac{1}{\sqrt{2\pi}\sigma_{il} y_i} \exp\left\{-\frac{1}{2}\frac{(\log(y_i) - \mu_{il})^2}{\sigma_{il}^2}\right\},$$

Weibull

$$f(y_i|\boldsymbol{\theta}_{il}) = \frac{\beta_{il}}{\theta_{il}} \left(\frac{y_i}{\theta_{il}}\right)^{\beta_{il}-1} \exp\left\{-\left(\frac{y_i}{\theta_{il}}\right)^{\beta_{il}}\right\},$$

gamma

$$f(y_i|\boldsymbol{\theta}_{il}) = \frac{1}{\Gamma[\beta_{il}] y_i} \left(\frac{y_i}{\theta_{il}}\right)^{\beta_{il}} \exp\left\{-\frac{y_i}{\theta_{il}}\right\},$$

binomial

$$f(y_i|\boldsymbol{\theta}_{il}) = \binom{\theta_{il}}{y_i} p_{il}^{y_i} (1 - p_{il})^{\theta_{il} - y_i},$$

Poisson

$$f(y_i|\boldsymbol{\theta}_{il}) = \frac{e^{-\theta_{il}} \theta_{il}^{y_i}}{y_i!}$$

or Dirac

$$f(y_i|\boldsymbol{\theta}_{il}) = \begin{cases} 1 & y_i = \theta_{il} \\ 0 & \text{otherwise} \end{cases}$$

parametric family types. The objective of the analysis is the inference about the number c of components, component weights w_l summing to 1 and component parameters $\boldsymbol{\theta}_l$. The REBMIX algorithm is an iterative numerical procedure relying on the suppositions:

- It is always possible to assign empirical densities to an arbitrary dataset.
- Based on the empirical densities, global mode position can be identified.
- Once the global mode position and its empirical density are known, rough component parameters of the predictive component density can be estimated.
- Based on the rough component parameters, the dataset can be clustered successively into the classes linked to the predictive component densities and the residue.
- The number c of components equals the number of the classes.
- Enhanced component parameters and the component weights can be assessed for all classes.
- The remaining observations can be distributed between the existing components by the Bayes decision rule and the parameters of the finite mixture can be fine-tuned.

Sections Preprocessing of observations to Bayes classification of the remaining observations give the theoretical backgrounds for the algorithm, while Section Algorithm flow lists and explains its flow.

2.1 Preprocessing of observations

The algorithm requires the preprocessing of observations. By the histogram approach, the dataset is counted into a finite number of nonoverlapping, equally sized and regularly distributed bins. Assuming that bin means $\bar{\boldsymbol{y}}_j = (\bar{y}_{1j}, \ldots, \bar{y}_{dj})^\top$ are given by

$$\bar{y}_{ij} = \bar{y}_{i0} + \text{'An arbitrary integer'} \times h_{ij}, \; i = 1, \ldots, d, \tag{3}$$

the fraction of observations k_j for $j = 1, \ldots, v$ falling into volume V_j is counted out, where $\bar{y}_{i0}$ stands for an arbitrary origin and v depicts the number of nonempty bins. Similarly, if the Parzen window is employed, the fraction of observations falling into V_j centered on observation $\boldsymbol{y}_j$ is obtained. In both cases, the volume is taken to be a hypersquare with the sides of length h_{ij}. This yields $V_j = \prod_{i=1}^{d} h_{ij}$. Moreover, for both approaches class widths $h_{ij} = h_i$ and volumes $V_j = V$ are kept constant. If the k-nearest neighbour approach is used, the fraction of observations falling into normalized hypersphere $V_j = \pi^{d/2} R_j^d / \Gamma[1 + d/2]$ of radius R_j centered on observation $\boldsymbol{y}_j$ contains constant number $k_j = k$ of observations. The class widths for the histogram and Parzen window approach and continuous parametric families

$$h_i = \frac{y_{i\max} - y_{i\min}}{v}$$

depend on the minimum $y_{i\min} = \min y_{ij}$ and maximum $y_{i\max} = \max y_{ij}$ observations. For the histogram approach and continuous parametric families origin is preset to

$$\bar{y}_{i0} = y_{i\min} + \frac{h_i}{2}.$$

However, discrete parametric families require $h_i = 1$ and $\bar{y}_{i0} = y_{i\min}$. The kth $- 1$ nearest neighbour $\boldsymbol{y}_{\hat{j}}$ is searched around $\boldsymbol{y}_j$ based on the normalized Euclidean distance

$$R_j = \sqrt{\sum_{i=1}^{d} \left(\frac{y_{i\hat{j}} - y_{ij}}{y_{i\max} - y_{i\min}} \right)^2} \text{ for } \hat{j} \neq j \text{ and } h_{ij} = 2R_j(y_{i\max} - y_{i\min}).$$

If $N \geq k$ nearest neighbours coincide, then the normalized Euclidean distance R_j to the first nearest non-coincident neighbour $\boldsymbol{y}_{\hat{j}}$ is multiplied by $(k/(N+1))^{1/d}$. Infinite empirical density estimations are thus prevented.

2.2 Global mode detection

Argument m at which empirical density f_{lj}

$$m = \arg\max_{j} f_{lj} \tag{4}$$

attains its maximum determines the global mode. If observations are binned into the histogram, then

$$f_{lj} = \frac{k_{lj}}{n_l}\frac{1}{V_j}, \ j = 1, \ldots, v, \tag{5}$$

where frequencies k_{lj} are all set to k_j initially and number of observations in class l is

$$n_l = \sum_{j=1}^{v} k_{lj}.$$

If the Parzen window or k-nearest neighbour approach is applied,

$$f_{lj} = \frac{k_{lj}}{n_l}\frac{k_j}{V_j}, \ j = 1, \ldots, n. \tag{6}$$

Frequencies k_{lj} are all set to 1 initially, $n_l = \sum_{j=1}^{n} k_{lj}$ and component weight $w_l = n_l/n$. Moreover, the lth component conditional empirical density at the global mode for the histogram approach

$$f_{i|\hat{\imath}.lm} = \frac{k_{lm}}{\sum_{j=1,\ \bar{y}_{\hat{\imath}j}=\bar{y}_{\hat{\imath}m}}^{v} k_{lj}}\frac{1}{h_{im}} = \frac{k_{lm}}{k_{i|\hat{\imath}.lm}}\frac{1}{h_{im}} \tag{7}$$

is required, where index $\hat{\imath} = 1, \ldots, i-1, i+1, \ldots, d$. If $d = 1$, then $k_{i|\hat{\imath}.lm} = n_l$ and $f_{i|\hat{\imath}.lm} = f_{lm}$. For the Parzen window and k-nearest neighbour approach

$$f_{i|\hat{\imath}.lm} = \frac{k_{lm}}{\sum_{j=1,\ |y_{\hat{\imath}j}-y_{\hat{\imath}m}|\leq h_{\hat{\imath}m}/2}^{n} k_{lj}}\frac{k_m}{h_{im}} = \frac{k_{lm}}{k_{i|\hat{\imath}.lm}}\frac{k_m}{h_{im}}. \tag{8}$$

2.3 Clustering of observations

The clustering of observations is an iterative procedure of identifying the observations belonging to the lth component. The deviations between k_{lj} and the predictive component frequencies for the histogram approach are given by

$$e_{lj} = k_{lj} - n_l f(\bar{\boldsymbol{y}}_j|\boldsymbol{\theta}_l)V_j. \tag{9}$$

For the Parzen window and k-nearest neighbour approach

$$e_{lj} = k_{lj} - n_l f(\boldsymbol{y}_j|\boldsymbol{\theta}_l)V_j/k_j. \tag{10}$$

To identify the most deviating observations, relative positive deviations $\varepsilon_{lj} = e_{lj}/k_{lj}$ and maximum positive relative deviation $\varepsilon_{l\max}$ are calculated. Total of positive and negative deviations

$$e_{l\mathrm{p}} = \sum_{j=1,\ e_{lj}>0}^{v} e_{lj} \text{ and } e_{l\mathrm{n}} = \sum_{j=1,\ e_{lj}<0}^{v} \max\{e_{lj}, -r_j\},$$

where r_j stands for the residual frequency. If index v is replaced by n the equation can be used with the Parzen window and k-nearest neighbour approach, too. Total of positive relative deviations of the lth component is then

$$D_l = \frac{e_{l\mathrm{p}}}{n_l}, \tag{11}$$

where $0 \leq D_l \leq 1$. The observations that inequality $\varepsilon_{lj} > \varepsilon_{l\max}(1 - a_{\mathrm{r}})$ holds for are not assumed to belong to the lth component and therefore move to the residue. Number of iterations depends on acceleration rate $0 < a_{\mathrm{r}} \leq 1$. It is best to keep a_{r} close to zero. The recommended value is 0.1. On the contrary, the observations where $e_{lj} < 0$ are transferred back to the lth component. The clustering of observations continues with the renewed rough parameter and component weight estimation until

$$D_l \leq \frac{D_{\min}}{w_l}. \tag{12}$$

Constant $0 < D_{\min} \leq 1$ is optimized by the information criterion. The clustering of observations ends with the enhanced component parameter estimation.

2.4 Rough component parameter estimation

The clustering of observations depends on the rough component parameters. Proper extraction of observations belonging to the lth component is assured by the restraints that prevent the component from its flowing away from the global mode as at least one component is supposed to be in its vicinity. This yields

$$f(\boldsymbol{y} = \hat{\boldsymbol{y}}_m | \boldsymbol{\theta}_l) = f_{lm}, \tag{13}$$

where $\hat{\boldsymbol{y}}_m = \bar{\boldsymbol{y}}_m$ for the histogram and $\hat{\boldsymbol{y}}_m = \boldsymbol{y}_m$ for the Parzen window and k-nearest neighbour approach. Restraint (13) is insufficient if $d > 1$ even for single parameter component densities, such as for Dirac and exponential. Allowing for the independence of components (2) equation (13) yields

$$\prod_{i=1}^{d} f(y_i = \hat{y}_{im} | \boldsymbol{\theta}_{il}) = f_{lm} = \prod_{i=1}^{d} \varepsilon f_{i|\hat{\imath}.lm}, \tag{14}$$

wherefrom required restraints

$$f(y_i = \hat{y}_{im} | \boldsymbol{\theta}_{il}) = \varepsilon f_{i|\hat{\imath}.lm} = f_{i|\hat{\imath}.l\max}, \; i = 1, \ldots, d \tag{15}$$

can be derived. In addition, from known f_{lm} and $f_{i|\hat{\imath}.lm}$ it follows

$$\varepsilon = \min \left\{ 1, \left(\frac{f_{lm}}{\prod_{i=1}^{d} f_{i|\hat{\imath}.lm}} \right)^{\frac{1}{d}} \right\}, \tag{16}$$

where the upper limit of ε is set to 1. For Rayleigh, Poisson or binomial distribution with known θ_{il} it is assumed

$$\frac{\partial f(y_i = \hat{y}_{im} | \boldsymbol{\theta}_{il})}{\partial y_i} = 0, \; i = 1, \ldots, d. \tag{17}$$

The rough component parameters for single parameter distributions are thus gained from (15) or (17). For two parameter normal, lognormal, Weibull or gamma distribution Lagrange multiplier

$$\Lambda(\boldsymbol{\theta}_{il}, \lambda_{il}) = -\int_{-\infty}^{+\infty} f(y_i | \boldsymbol{\theta}_{il}) \log(f(y_i | \boldsymbol{\theta}_{il})) dy_i + \lambda_{il} \log(f(y_i = \hat{y}_{im} | \boldsymbol{\theta}_{il}) / f_{i|\hat{\imath}.l\max}) \tag{18}$$

provides a strategy for entropy maximization subject to logarithm of (15). The rough component parameters for two parameter distributions are then a solution of

$$\nabla_{\boldsymbol{\theta}_{il}, \lambda_{il}} \Lambda(\boldsymbol{\theta}_{il}, \lambda_{il}) = 0, \; i = 1, \ldots, d. \tag{19}$$

Constrained entropy (18) maximization enables rough Weibull and gamma parameter estimation for shape parameter $\beta_{il} > 0$ and not only for $\beta_{il} > 1$ as in Nagode and Fajdiga (2011a,b). Rough normal component parameters are given by

$$\mu_{il} = \hat{y}_{im} \text{ and } \sigma_{il} = \frac{1}{\sqrt{2\pi} f_{i|\hat{\imath}.l\max}}. \tag{20}$$

Similarly, rough lognormal

$$f(\lambda_{il}) = \frac{\lambda_{il} - 1}{\lambda_{il}} + \log(\lambda_{il}(\lambda_{il} - 1)) + 2\log(\sqrt{2\pi} f_{i|\hat{\imath}.l\max}\hat{y}_{im}) = 0,$$

$$\mu_{il} = \lambda_{il} - 1 + \log(\hat{y}_{im}) \text{ and } \sigma_{il} = \sqrt{\lambda_{il}(\lambda_{il} - 1)}, \quad (21)$$

Weibull

$$f(\alpha_{il}) = \frac{\alpha_{il} - 1}{\lambda_{il}} e^{\frac{1}{\alpha_{il}}} - f_{i|\hat{\imath}.l\max}\hat{y}_{im} e = 0, \; \lambda_{il} = \frac{\alpha_{il}}{\beta_{il}},$$

$$\beta_{il} = \alpha_{il} + \gamma + \log\left(\frac{\alpha_{il} - 1}{\alpha_{il}}\right), \; \theta_{il} = \hat{y}_{im}\left(\frac{\alpha_{il}}{\alpha_{il} - 1}\right)^{\frac{1}{\beta_{il}}} \text{ and } \beta_{il} > 0, \quad (22)$$

gamma

$$f(\alpha_{il}) = \frac{1}{2}\log(\beta_{il}) + \beta_{il}\left(\log\left(\frac{\alpha_{il} - 1}{\alpha_{il}}\right) + \frac{1}{\alpha_{il}}\right) - \log(\sqrt{2\pi} f_{i|\hat{\imath}.l\max}\hat{y}_{im}) = 0,$$

$$\beta_{il} = \frac{\gamma(1 + \alpha_{il})}{\gamma - 1 - \alpha_{il}\log\left(\frac{\alpha_{il} - 1}{\alpha_{il}}\right)}, \; \lambda_{il} = \frac{\alpha_{il}}{\beta_{il}}, \; \theta_{il} = \frac{\hat{y}_{im}\lambda_{il}}{\alpha_{il} - 1} \text{ and } \beta_{il} > 0, \quad (23)$$

binomial

$$p_{il} = \begin{cases} 1 - f_{i|\hat{\imath}.l\max}^{1/\theta_{il}} & \hat{y}_{im} = 0 \\ f_{i|\hat{\imath}.l\max}^{1/\theta_{il}} & \hat{y}_{im} = \theta_{il} \\ \hat{y}_{im}/\theta_{il} & \text{otherwise}, \end{cases} \quad (24)$$

rough Poisson

$$\theta_{il} = \begin{cases} -\log(f_{i|\hat{\imath}.l\max}) & \hat{y}_{im} = 0 \\ \hat{y}_{im} & \text{otherwise} \end{cases} \quad (25)$$

and rough Dirac

$$\theta_{il} = \hat{y}_{im} \quad (26)$$

component parameters are derived, where γ is the Euler-Mascheroni constant. When deriving (23) $\Gamma[\beta_{il}]$ is approximated by the Stirling's formula and digamma function by $\psi(\beta_{il}) = \log(\beta_{il}) - \gamma/\beta_{il}$. Rough binomial parameter $\theta_{il} = \theta_i$ is fixed and equals the number of categories minus one.

The rigid restraints result in poor component parameter estimation if modes of several component densities coincide. The loose restraints introduced in Nagode and Fajdiga (2011a) improve component parameter estimation and offer further evolution opportunities. The rigid restraints become loose if $f_{i|\hat{\imath}.l\max}$ in equations (20) to (26) is replaced by $f_{i|\hat{\imath}.lm}$, where

$$0 \leq f_{i|\hat{\imath}.lm} \leq f_{i|\hat{\imath}.l\max}. \quad (27)$$

Instead of minimizing the maximum relative positive deviation (Nagode and Fajdiga, 2011a) the simpler root finding of the total of relative deviations is used here to attain the optimal $f_{i|\hat{\imath}.lm}$. For the histogram approach total of relative deviations

$$D_{i|\hat{\imath}.lm} = 1 - \sum_{j=1,\; \bar{y}_{\hat{\imath}j} = \bar{y}_{\hat{\imath}m}}^{v} f(y_i = \bar{y}_{ij}|\boldsymbol{\theta}_{il})h_{ij} \quad (28)$$

equals the fraction of observations falling into the regions on y_i axis with zero empirical probability. If $D_{i|\hat{\imath}.lm}$ is close to zero, e.g., 0.002, then observations not contributing significantly to the lth component should not affect the loose component parameter estimation. This yields

$$\sum_{j=1,\; \bar{y}_{\hat{\imath}j} = \bar{y}_{\hat{\imath}m}}^{v} f(\bar{y}_{ij}|\boldsymbol{\theta}_{il})h_{ij} = 0.998 \quad (29)$$

Equation (29) can be solved for optimal $f_{i|\hat{i}.lm}$ by the bisection root finding method. If the root does not exist, then $f_{i|\hat{i}.lm} = f_{i|\hat{i}.l\max}$. For the Parzen window and k-nearest neighbour approach the root of

$$\sum_{j=1,\ |y_{\hat{i}j}-y_{\hat{i}m}|\leq h_{\hat{i}m}/2}^{n} f(y_{ij}|\boldsymbol{\theta}_{il})h_{ij}/k_j = 0.998 \tag{30}$$

is searched for optimal $f_{i|\hat{i}.lm}$. Dirac parameter θ_{il} of (26) does not require $f_{i|\hat{i}.lm}$ optimization.

2.5 Enhanced component parameter estimation

Maximum likelihood is employed to get enhanced component parameters. For the histogram approach enhanced normal component parameters are given by

$$\mu_{il} = \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\hat{y}_{ij} \text{ and } \sigma_{il}^2 = \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\hat{y}_{ij}^2 - \mu_{il}^2. \tag{31}$$

Likewise, enhanced lognormal

$$\mu_{il} = \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\log(\hat{y}_{ij}) \text{ and } \sigma_{il}^2 = \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\log(\hat{y}_{ij})^2 - \mu_{il}^2, \tag{32}$$

Weibull

$$\theta_{il}^{\beta_{il}} = \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\hat{y}_{ij}^{\beta_{il}} \text{ and } f(\beta_{il}) = \frac{1}{\beta_{il}} + \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\log(\hat{y}_{ij}) - \frac{\sum_{j=1}^{v} k_{lj}\hat{y}_{ij}^{\beta_{il}}\log(\hat{y}_{ij})}{\sum_{j=1}^{v} k_{lj}\hat{y}_{ij}^{\beta_{il}}} = 0, \tag{33}$$

gamma

$$\theta_{il} = \frac{1}{\beta_{il}n_l}\sum_{j=1}^{v} k_{lj}\hat{y}_{ij} \text{ and } f(\beta_{il}) = \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\log(\hat{y}_{ij}) - \log(\theta_{il}) - \frac{\Gamma'[\beta_{il}]}{\Gamma[\beta_{il}]} = 0, \tag{34}$$

binomial

$$p_{il} = \frac{1}{n_l\theta_{il}}\sum_{j=1}^{v} k_{lj}\hat{y}_{ij}, \tag{35}$$

Poisson

$$\theta_{il} = \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\hat{y}_{ij} \tag{36}$$

and Dirac component parameters

$$\theta_{il} = \hat{y}_{im} \tag{37}$$

are estimated. Index v is replaced by n for the Parzen window or k-nearest neighbour approach.

2.6 First and second moment calculation

The first and second moment of the normal

$$m_{il} = \mu_{il} \text{ and } V_{il} = \sigma_{il}^2 + \mu_{il}^2, \tag{38}$$

lognormal

$$m_{il} = e^{\mu_{il}+\frac{\sigma_{il}^2}{2}} \text{ and } V_{il} = e^{2\mu_{il}+2\sigma_{il}^2}, \tag{39}$$

Weibull

$$m_{il} = \theta_{il}\Gamma\left[1+\frac{1}{\beta_{il}}\right] \text{ and } V_{il} = \theta_{il}^2\Gamma\left[1+\frac{2}{\beta_{il}}\right], \tag{40}$$

gamma

$$m_{il} = \theta_{il}\beta_{il} \text{ and } V_{il} = \theta_{il}^2\beta_{il}(1+\beta_{il}) \tag{41}$$

and the first moment of binomial

$$m_{il} = \theta_{il} p_{il}, \tag{42}$$

Poisson

$$m_{il} = \theta_{il} \tag{43}$$

and Dirac

$$m_{il} = \theta_{il} \tag{44}$$

distributions are calculated to enable the classification of the remaining observations.

2.7 Bayes classification of the remaining observations

With the increase of the number of components, the number n_l of the remaining observations decreases. When the component weight attains the minimum weight

$$w_l \leq w_{\min} = 2D_{\min}(l-1) \tag{45}$$

it is assumed that remaining observations k_{lj} belong to the existing classes and do not form the new ones. The classification of the remaining observations is accomplished by the Bayes decision rule (Duda and Hart, 1973)

$$\begin{gathered} l = \arg\max_l w_l f(\boldsymbol{y}_j|\boldsymbol{\theta}_l) \\ w_l = w_l + \frac{k_{lj}}{n}, \ m_{il} = m_{il} + \frac{k_{lj}(y_{ij} - m_{il})}{nw_l} \text{ and } V_{il} = V_{il} + \frac{k_{lj}(y_{ij}^2 - V_{il})}{nw_l}, \end{gathered} \tag{46}$$

where k_{lj} is added to the lth class and the component weight and both moments are recalculated (Bishop, 1995). Once all v bin means or all n observations are processed, the predictive mixture parameters are gained by inverting (38) to (44).

2.8 Algorithm flow

The REBMIX is listed in Figure 1. It requires fifteen arguments, whereby depending on the parametric families five or six of them are mandatory, the rest is optional. It consists of three main loops: the inner 9 → 37, the middle 6 → 41 and the outer loop 4 → 47. The numbers are line indices. In line 2 the observations are preprocessed as described in Section Preprocessing of observations. In line 3, counter I_1, constant $D_{\min}$ and frequencies k_{lj} are initiated. Next, the outer loop begins. Line 5 presumes that the mixture consists of one component, then the number r of observations to separate is set to n and n_l to n. If ratio n_l/n is greater than the minimum weight introduced in Section Bayes classification of the remaining observations, the middle loop enters. Otherwise, the finite mixture parameter estimation for $v \in K$ is completed.

In lines 7 and 8, global mode argument m is detected as explained in Section Global mode detection, counter I_2 is initiated, component weight w_l is calculated and frequencies r_j are all set to zero. If $I_2 \leq I_{\max}$, the inner loop enters, otherwise in line 38 the first and the second moments are calculated (see Section First and second moment calculation). Next, number of components c is set to l, number of observations r is decreased by n_l, l is incremented, number r of the remaining observations joins n_l, residue frequencies r_j are all moved to k_{lj}, and the Stop criterion is determined.

The inner loop is divided into three sections. In line 10 the component parameters are estimated roughly (see Section Rough component parameter estimation). In the second section 11 → 23, total of positive relative deviations D_l and maximum relative deviation $\varepsilon_{l\max}$ are calculated. The number of iterations depends on acceleration rate a_{r}. In the third section 24 → 35, the maximum and negative deviations are transferred between frequencies k_{lj} and residue r_j. This way deviations e_{lj} are reduced gradually. The negative value of e_{lj} can never be higher than residue value r_j. If this is not true, deviation e_{lj} is corrected as listed in line 19. When the condition in line 24 is not fulfilled, the enhanced component parameter estimation is carried out (see Section Enhanced component parameter estimation) and the inner loop ends.

```
Require: Dataset[1], Preprocessing[1], D, cmax, Criterion, Variables[1], pdf[1], Theta1[1], Theta2, K[1], y0, ymin, ymax, ar and
    Restraints.
Ensure: Dataset contains datasets, Preprocessing is one of "histogram", "Parzen window" or "k-nearest neighbour", $0 \le$
    $\texttt{D} \le 1$, $\texttt{cmax} \in \mathbb{N}$, Criterion is one of "AIC", "AIC3", "AIC4", "AICc", "BIC", "CAIC", "HQC", "MDL2", "MDL5", "AWE", "CLC",
    "ICL", "PC", "ICL-BIC", "D" or "SSE", Variables are "continuous" or "discrete", pdf is one of "normal", "lognormal",
    "Weibull", "gamma", "binomial", "Poisson" or "Dirac", Theta1 may contain initial binomial parameters, Theta2 is inactive,
    $\texttt{K} \subset \mathbb{N}$, y0 may contain origins, ymin and ymax may contain minimum and maximum observations, $0 < \texttt{ar} \le 1$ and Restraints
    are "loose" or "rigid".
for all $v$ such that $v \in \texttt{K}$ do
    Preprocessing of observations
    $I_1 \leftarrow 1$, $D_{\min} \leftarrow 0.25$, $k_{lj} \leftarrow k_j$ for $j = 1$ to $v$
    repeat
        $l \leftarrow 1$, $r \leftarrow n$, $n_l \leftarrow n$
        while $n_l/n > 2D_{\min}(l-1)$ do
            Global mode detection
            $I_2 \leftarrow 1$, $w_l \leftarrow n_l/n$, $r_j \leftarrow 0$ for $j = 1$ to $v$
            while $I_2 \le I_{\max}$ do
                Rough component parameter estimation
                $e_{l\mathrm{p}} \leftarrow 0$, $e_{l\mathrm{n}} \leftarrow 0$, $e_{l\max} \leftarrow 0$
                for $j = 1$ to $v$ do
                    $e_{lj} \leftarrow 0$, $\varepsilon_{lj} \leftarrow 0$
                    if $k_{lj} > 0$ or $r_j > 0$ then
                        $e_{lj} \leftarrow k_{lj} - n_l f(\bar{\boldsymbol{y}}_j|\boldsymbol{\theta}_l)V_j$
                        if $e_{lj} > 0$ then
                            $\varepsilon_{lj} \leftarrow e_{lj}/k_{lj}$, $\varepsilon_{l\max} \leftarrow \max\{\varepsilon_{l\max}, \varepsilon_{lj}\}$, $e_{l\mathrm{p}} \leftarrow e_{l\mathrm{p}} + e_{lj}$
                        else
                            $e_{lj} \leftarrow \max\{e_{lj}, -r_j\}$, $e_{l\mathrm{n}} \leftarrow e_{l\mathrm{n}} - e_{lj}$
                        end if
                    end if
                end for
                $D_l \leftarrow e_{l\mathrm{p}}/n_l$, $\varepsilon_{l\max} \leftarrow \varepsilon_{l\max}(1 - \texttt{ar})$
                if $D_l > D_{\min}/w_l$ then
                    for all $j$ such that $1 \le j \le v$ and $\varepsilon_{lj} > \varepsilon_{l\max}$ do
                        $k_{lj} \leftarrow k_{lj} - e_{lj}$, $r_j \leftarrow r_j + e_{lj}$, $n_l \leftarrow n_l - e_{lj}$
                    end for
                    $e_{l\mathrm{p}} \leftarrow e_{l\mathrm{p}}/D_l - n_l$, $f \leftarrow e_{l\mathrm{p}}/e_{l\mathrm{n}}$ if $e_{l\mathrm{n}} > e_{l\mathrm{p}}$ otherwise $f \leftarrow 1$
                    for all $j$ such that $1 \le j \le v$ and $e_{lj} < 0$ do
                        $e_{lj} \leftarrow f e_{lj}$, $k_{lj} \leftarrow k_{lj} - e_{lj}$, $r_j \leftarrow r_j + e_{lj}$, $n_l \leftarrow n_l - e_{lj}$
                    end for
                    $w_l \leftarrow n_l/n$
                else
                    Enhanced component parameter estimation, break
                end if
                $I_2 \leftarrow I_2 + 1$
            end while
            First and second moment calculation
            $c \leftarrow l$, $r \leftarrow r - n_l$, $l \leftarrow l + 1$, $n_l \leftarrow r$, $k_{lj} \leftarrow r_j$ for $j = 1$ to $v$
            Stop $\leftarrow c \ge v$ or $c \ge \texttt{cmax}$, break if Stop = true
        end while
        Bayes classification of the remaining observations, log likelihood $\log L$, information criterion IC and total of positive
            relative deviations $D$ calculation
        if IC < $\mathrm{IC}_{\mathrm{opt}}$ then
            $\log L \rightarrow \log L_{\mathrm{opt}}$, $\mathrm{IC} \rightarrow \mathrm{IC}_{\mathrm{opt}}$, $c \rightarrow c_{\mathrm{opt}}$, $\boldsymbol{w} \rightarrow \boldsymbol{w}_{\mathrm{opt}}$, $\boldsymbol{\Theta} \rightarrow \boldsymbol{\Theta}_{\mathrm{opt}}$
        end if
        Stop $\leftarrow$ Stop or $D \le \texttt{D}$ or $I_1 \ge I_{\max}$, $D_{\min} \leftarrow cD_{\min}/(c+1)$, $I_1 \leftarrow I_1 + 1$
    until Stop = true
end for
```

Figure 1: REBMIX algorithm.

The enhanced component parameter estimation may fail. In this instance, the component parameters are reset to the state just before the failure occurred. In line 42 the remaining observations are classified by the Bayes decision rule as depicted in Section Bayes classification of the remaining observations. Further on, information criterion, e.g., Akaike (1974)

$$\mathrm{IC} = -2 \log L(c, \boldsymbol{w}, \boldsymbol{\Theta}) + 2M \tag{47}$$

is calculated, whereas the number of free parameters for the normal, lognormal, Weibull and gamma mixtures can be written as

$$M = 2cd + c - 1. \tag{48}$$

The binomial, Poisson and Dirac mixtures require $M = cd + c - 1$. The log likelihood function for the binned observations is given by

$$\log L(c, \boldsymbol{w}, \boldsymbol{\Theta}) = \sum_{j=1}^{v} k_j \log f(\bar{\boldsymbol{y}}_j | c, \boldsymbol{w}, \boldsymbol{\Theta}). \tag{49}$$

Otherwise,

$$\log L(c, \boldsymbol{w}, \boldsymbol{\Theta}) = \sum_{j=1}^{n} \log f(\boldsymbol{y}_j | c, \boldsymbol{w}, \boldsymbol{\Theta}). \tag{50}$$

Finally, total of positive relative deviations for the histogram

$$D = \sum_{j=1}^{v} \left\langle \frac{k_j}{n} - f(\bar{\boldsymbol{y}}_j | c, \boldsymbol{w}, \boldsymbol{\Theta}) V_j \right\rangle, \tag{51}$$

Parzen window or k-nearest neighbour

$$D = \sum_{j=1}^{n} \left\langle \frac{1}{n} - \frac{f(\boldsymbol{y}_j | c, \boldsymbol{w}, \boldsymbol{\Theta}) V_j}{k_j} \right\rangle \tag{52}$$

is calculated, where $\langle x \rangle = x$ if $x > 0$ and $\langle x \rangle = 0$ if $x \leq 0$. This way global optimum $\mathrm{IC}_{\mathrm{opt}}$ corresponding to the optimal number c_{opt} of components, weights $\boldsymbol{w}_{\mathrm{opt}}$ and parameters $\boldsymbol{\Theta}_{\mathrm{opt}}$ can always be found. In line 46, the Stop criterion is redetermined and D_{min} is decreased in such a way that total of positive relative deviations

$$c D_{\mathrm{min}}^{\mathrm{old}} = (c + 1) D_{\mathrm{min}}^{\mathrm{new}}$$

for c and $c + 1$ components is preserved. When line 47 is fulfilled, the procedure stops. If index v in Figure 1 is replaced by n and line 15 is replaced by (10) the algorithm, presented for the histogram approach, can also be used with the Parzen window and k-nearest neighbour approach.

3 Examples

To illustrate the use of the REBMIX algorithm, univariate and multivariate datasets are considered. The **rebmix** is loaded and the prompt before starting new page is set to `TRUE`.

```
R> library("rebmix")
R> devAskNewPage(ask = TRUE)
```

3.1 Gamma datasets

Three gamma mixtures are considered (Wiper et al., 2001). The first has four well-separated components with means 2, 4, 6 and 8, respectively

$$\begin{array}{lll} \theta_1 = 1/100 & \beta_1 = 200 & n_1 = 100 \\ \theta_2 = 1/100 & \beta_2 = 400 & n_2 = 100 \\ \theta_3 = 1/100 & \beta_3 = 600 & n_3 = 100 \\ \theta_4 = 1/100 & \beta_4 = 800 & n_4 = 100. \end{array}$$

The second has equal means but different variances and weights

$$\begin{array}{lll} \theta_1 = 1/27 & \beta_1 = 9 & n_1 = 40 \\ \theta_2 = 1/270 & \beta_2 = 90 & n_2 = 360. \end{array}$$

The third is a mixture of a rather diffuse component with mean 6 and two lower weighted components with smaller variances and means of 2 and 10, respectively

$$\begin{array}{lll} \theta_1 = 1/20 & \beta_1 = 40 & n_1 = 80 \\ \theta_2 = 1 & \beta_2 = 6 & n_2 = 240 \\ \theta_3 = 1/20 & \beta_3 = 200 & n_3 = 80. \end{array}$$

The gamma mixtures are generated by calling the `RNGMIX` function. It demands character vector `Dataset` containing list names of data frames that datasets are written in, random seed `rseed`, vector `n` containing number of observations in classes n_l and a matrix containing c parametric family types `pdfi`. One of `"normal"`, `"lognormal"`, `"Weibull"`, `"gamma"`, `"binomial"`, `"Poisson"` or `"Dirac"`. Component parameters `theta1.i` follow the parametric family types. One of μ_{il} for normal and lognormal distributions and θ_{il} for Weibull, gamma, binomial, Poisson and Dirac distributions. Component parameters `theta2.i` follow `theta1.i`. One of σ_{il} for normal and lognormal distributions, β_{il} for Weibull and gamma distributions and p_{il} for binomial distribution.

```
R> n <- c(100, 100, 100, 100)
R> Theta <- rbind(pdf = "gamma", theta1 = c(1/100, 1/100, 1/100,
+      1/100), theta2 = c(200, 400, 600, 800))
R> gamma1 <- RNGMIX(Dataset = "gamma1", n = n, Theta = Theta)
R> n <- c(40, 360)
R> Theta <- rbind(pdf = "gamma", theta1 = c(1/27, 1/270), theta2 = c(9,
+      90))
R> gamma2 <- RNGMIX(Dataset = "gamma2", n = n, Theta = Theta)
R> n <- c(80, 240, 80)
R> Theta <- rbind(pdf = "gamma", theta1 = c(1/20, 1, 1/20), theta2 = c(40,
+      6, 200))
R> gamma3 <- RNGMIX(Dataset = "gamma3 ", n = n, Theta = Theta)
```

The `gamma1$Dataset`, `gamma2$Dataset` and `gamma3$Dataset` hold a list of data frames of size $n \times d$. See `help("RNGMIX")` in **rebmix** for details. The preprocessing is set to histogram, maximum number of components to 8 and information criterion to AIC or BIC. The number of classes ranges from 30 to 80 and function `REBMIX` is called for the gamma parametric family type.

```
R> gamma1est <- REBMIX(Dataset = gamma1$Dataset, Preprocessing = "histogram",
+      cmax = 8, Criterion = c("AIC", "BIC"), Variables = "continuous",
+      pdf = "gamma", K = 30:80)
R> gamma2est <- REBMIX(Dataset = gamma2$Dataset, Preprocessing = "histogram",
+      cmax = 8, Criterion = "BIC", Variables = "continuous", pdf = "gamma",
+      K = 30:80)
R> gamma3est <- REBMIX(Dataset = gamma3$Dataset, Preprocessing = "histogram",
+      cmax = 8, Criterion = "BIC", Variables = "continuous", pdf = "gamma",
+      K = 30:80)
```

See `help("REBMIX")` in **rebmix** for details about specifying arguments for the function returning an object of class `REBMIX`. List of data frames `w` contains component weights w_l summing to 1, `Theta` stands for a list of data frames containing parametric family types `pdfi`. One of `"normal"`, `"lognormal"`, `"Weibull"`, `"gamma"`, `"binomial"`, `"Poisson"` or `"Dirac"`. Component parameters `theta1.i` follow the parametric family types. One of μ_{il} for normal and lognormal distributions and θ_{il} for Weibull, gamma, binomial, Poisson and Dirac distributions. Component parameters `theta2.i` follow `theta1.i`. One of σ_{il} for normal and lognormal distributions, β_{il} for Weibull and gamma distributions and p_{il} for binomial distribution. Character vector `Variables` contains types of variables. One of `"continuous"` or `"discrete"`.

In the `summary` data frame additional information about dataset, preprocessing, D, $c_{\max}$, information criterion type, a_{r}, restraints type, optimal c, optimal k, $\bar{y}_{i0}$, optimal h_i, information criterion IC and log likelihood $\log L$ is stored. Position `pos` in the `summary` data frame at which log likelihood $\log L$ attains its maximum is available, too. See `help("summary.REBMIX")` for details.

```
R> summary(gamma1est)
```

```
  Dataset Preprocessing Criterion c v/k IC   logL M
1 gamma1  histogram     AIC       5 59  1004 -488 14
2 gamma1  histogram     BIC       4 79  1054 -494 11
Maximum logL = -488 at pos = 1.
```

The `plot` method delivers fitted finite mixture with the legend in Figure 2. The corresponding predictive gamma mixture parameters are given by the `coef` method.

```
R> plot(gamma2est, pos = 1, what = c("den", "dis"), ncol = 2, npts = 1000)
```

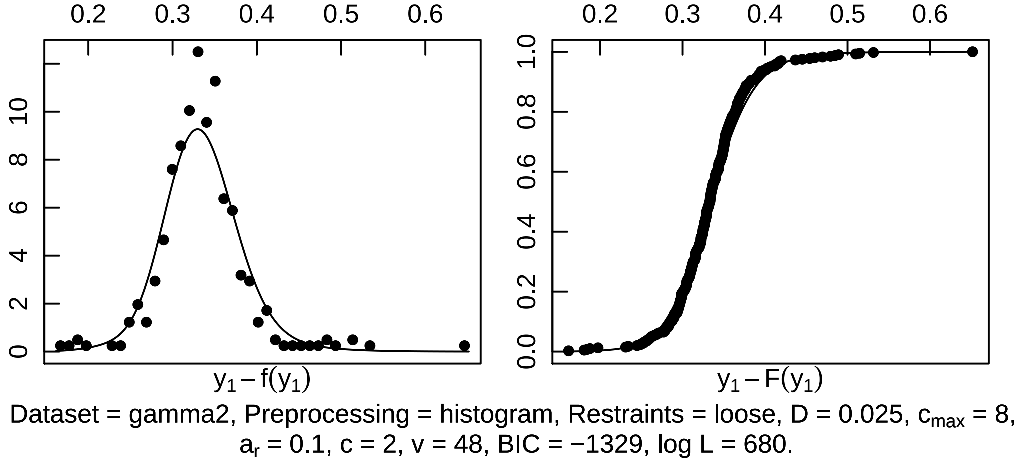

Figure 2: Gamma 2 dataset. Empirical density (circles) and predictive gamma mixture density in black solid line.

```
R> coef(gamma2est)
```

```
       comp1   comp2
w      0.824   0.176
pdf    gamma   gamma
theta1 0.00472 0.0166
theta2 71      20.2
```

For the details about specifying arguments for the `plot` and `coef` methods see `help("plot.REBMIX")` and `help("coef.REBMIX")`, respectively.

By calling the `boot.REBMIX` method `B` bootstrap datasets of length `n` are generated for the `x` object of class `REBMIX` at position `pos`, where bootstrap `Bootstrap` can be one of default `"parametric"` or `"nonparametric"`. Arguments `replace` and `prob` affect the nonparametric bootstrap only, see `help("sample")` and McLachlan and Peel (1997) for details about replacement and weighted bootstrap.

```
R> gamma3boot <- boot.REBMIX(x = gamma3est, pos = 1, Bootstrap = "p",
+      B = 10, n = NULL, replace = TRUE, prob = NULL)
```

```
R> gamma3boot
```

```
$c
 [1] 3 3 3 3 3 3 3 4 3 4

$c.mode
[1] 3

$c.prob
[1] 0.8

$c.se
[1] 0.422

$theta1.se
[1] 0.0359 0.2218 0.1666

$theta2.se
[1] 119 143 172

$w.se
[1] 0.0223 0.0904 0.0947

$c.cv
[1] 0.132

$theta1.cv
[1] 0.369 0.942 0.715

$theta2.cv
[1] 1.87 1.15 1.31

$w.cv
[1] 0.0769 0.2476 0.2745

attr(,"class")
[1] "boot.REBMIX"
```

The `gamma3boot` object of class `boot.REBMIX` holds a data frame `c` containing numbers c of components for B bootstrap datasets, standard error `c.se`, coefficient of variation `c.cv`, mode `c.mode` and mode probability `c.prob` of the numbers of components. Component weights `w`, component parameters `theta1.i` and `theta2.i`, standard errors `w.se`, `theta1.i.se` and `theta2.i.se` and coefficients of variation `w.cv`, `theta1.i.cv` and `theta2.i.cv` for those bootstrap datasets for which c equals mode c_m are returned, too. See `help("boot.REBMIX")` in **rebmix** for details.

```
R> summary(gamma3boot)
```

```
          comp1  comp2 comp3
w.cv      0.0769 0.248 0.274
theta1.cv 0.369  0.942 0.715
theta2.cv 1.87   1.15  1.31
Mode probability = 0.8 at c = 3 components.
```

3.2 Poisson dataset

Dataset consists of $n = 600$ two dimensional observations obtained by generating data points separately from each of three Poisson distributions. The component dataset sizes and parameters, which are those studied in Ma et al. (2009), are displayed below

$$\begin{array}{ll} \boldsymbol{\theta}_1 = (3, 2)^\top & n_1 = 200 \\ \boldsymbol{\theta}_2 = (9, 10)^\top & n_2 = 200 \\ \boldsymbol{\theta}_3 = (15, 16)^\top & n_3 = 200 \end{array}$$

For the dataset Ma et al. (2009) conduct 100 experiments by selecting different initial values of the mixing proportions. In all the cases, the adaptive gradient BYY learning algorithm leads to the correct model selection, i.e., finally allocating the correct number of Poissons for the dataset. In the meantime, it also results in an estimate for each parameter in the original or true Poisson mixture which generated the dataset. As the dataset of Ma et al. (2009) can not exactly be reproduced, 100 datasets are generated with random seeds r_{seed} ranging from -1 to -100.

```
R> n <- c(200, 200, 200)
R> Theta <- rbind(rep("Poisson", 3), c(3, 9, 15), rep("Poisson",
+     3), c(2, 10, 16))
R> poisson <- RNGMIX(Dataset = paste("Poisson_", 1:100, sep = ""),
+     n = n, Theta = Theta)
```

In total, 100 finite mixture estimations are performed by calling the `REBMIX` function.

```
R> poissonest <- REBMIX(Dataset = poisson$Dataset, Preprocessing = "histogram",
+     cmax = 6, Criterion = "MDL5", Variables = rep("discrete",
+         2), pdf = rep("Poisson", 2), K = 1)
R> c <- as.numeric(poissonest$summary$c)
R> IC <- as.numeric(poissonest$summary$IC)
```

The results are as follows:

```
R> summary(c)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2.00    2.00    3.00    2.89    4.00    4.00

R> summary(IC, digits = 5)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   6923    7098    7140    7148    7199    7322
```

The `REBMIX` function predicts 2.89 components on average, where probability of identifying exactly $c = 3$ components equals 0.31. To plot the mixture in Figure 3 the `plot` method is called.

4 Summary

The article presents the REBMIX algorithm and the **rebmix** package. Four datasets are studied on the x64 architecture. By applying the **tikzDevice** package (Sharpsteen and Bracken, 2013), LaTeX plots with legends can be obtained. The REBMIX algorithm can be used to assess the initial set of the unknown parameters and number of components for, e.g., the EM algorithm or as a standalone procedure that is a good compromise between the nonparametric and parametric methods to the finite mixture estimation. Its major advantages are robustness and time efficiency especially with the histogram preprocessing for all datasets sizes. The Parzen window and k-nearest neighbour preprocessing are more suitable for smaller datasets. Its advantages are more stressed if mixtures are composed of larger number of components. The **rebmix** package can be broadened to other parametric family types. The `predict` method that enables class membership prediction is available in the **rebmix** package, too. See `help("predict.list")` for details. The REBMIX can thus also be used for pattern recognition.

```
R> plot(poissonest, pos = 58, what = c("dens", "marg", "IC", "D",
+      "logL"), nrow = 2, ncol = 3, npts = 1000)
```

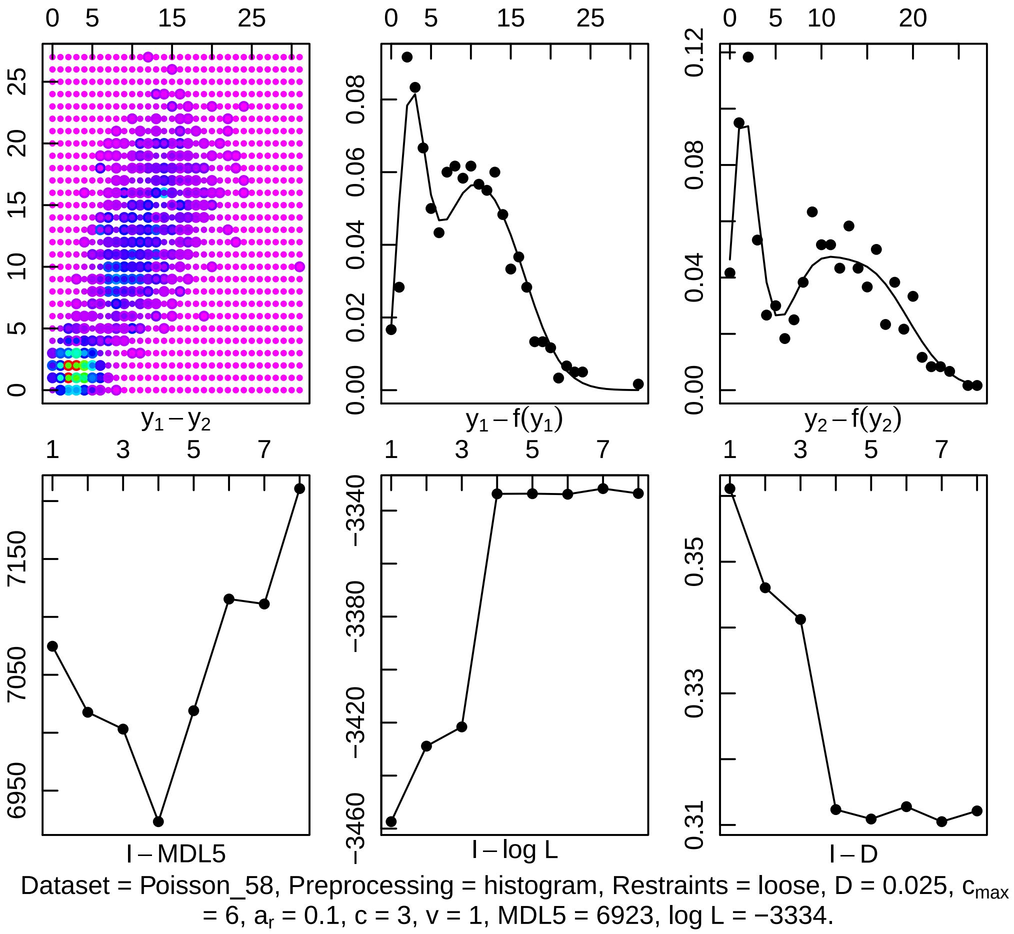

Figure 3: Poisson dataset. Empirical densities (coloured large circles), predictive multivariate Poisson-Poisson mixture density (coloured small circles), empirical densities (circles), predictive univariate marginal Poisson mixture densities and progress charts (solid line).

References

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(51):716–723, 1974.

D. Ardia, L. F. Hoogerheide, and H. K. van Dijk. Adaptive mixture of student-t distributions as a flexible candidate distribution for efficient simulation: The r package admit. *Journal of Statistical Software*, 29(3):1–31, 2009. URL `http://www.jstatsoft.org/v29/i03`.

T. Benaglia, D. Chauveau, D. R. Hunter, and D. S. Young. mixtools: An r package for analyzing finite mixture models. *Journal of Statistical Software*, 32(6):1–29, 2009. URL `http://www.jstatsoft.org/v32/i06`.

C. M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, 1995.

T. Bučar, M. Nagode, and M. Fajdiga. Reliability approximation using finite weibull mixture distributions. *Reliability Engineering & System Safety*, 84(3):241–251, 2004. doi: 10.1016/j.ress.2003.11.008.

G. Celeux, S. Chrétien, and F. Forbes. A component-wise em algorithm for mixtures. *Journal of Computational and Graphical Statistics*, 10(4):697–712, 2001. URL `http://www.jstor.org/stable/1390967`.

M. Chavent, V. Kuentz-Simonet, B. Liquet, and J. Saracco. Clustofvar: An r package for the clustering of variables. *Journal of Statistical Software*, 50(13):1–16, 2012. URL `http://www.jstatsoft.org/v50/i13`.

M. Chen, M. Gunel, and H. Zhao. Somatica: Identifying, characterizing and quantifying somatic copy number aberrations from cancer genome sequencing data. *PLOS ONE*, 8(11):1–10, 2013. doi: 10.1371/journal.pone.0078143.

R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, New York, 1973.

C. Fraley and A. Raftery. Model-based methods of classification: Using the mclust software in chemometrics. *Journal of Statistical Software*, 18(6):1–13, 2007. URL `http://www.jstatsoft.org/v18/i06`.

S. Frühwirth-Schnatter. *Finite Mixture and Markov Switching Models.* Springer-Verlag Series in Statistics, New York, 2006.

B. Grün and F. Leisch. Flexmix version 2: Finite mixtures with concomitant variables and varying and constant parameters. *Journal of Statistical Software*, 28(4):1–35, 2008. URL `http://www.jstatsoft.org/v28/i04`.

B. Grün, I. Kosmidis, and A. Zeileis. Extended beta regression in r: Shaken, stirred, mixed, and partitioned. *Journal of Statistical Software*, 48(11):1–25, 2012. URL `http://www.jstatsoft.org/v48/i11`.

S. Ingrassiaa and R. Roccib. Constrained monotone em algorithms for finite mixture of multivariate gaussians. *Computational Statistics & Data Analysis*, 51(11):5339–5351, 2007. doi: 10.1016/j.csda.2006.10.011.

F. Leisch. Flexmix: A general framework for finite mixture models and latent class regression in r. *Journal of Statistical Software*, 11(8):1–18, 2004. URL `http://www.jstatsoft.org/v11/i08`.

J. Ma, J. Liu, and Z. Ren. Parameter estimation of poisson mixture with automated model selection through byy harmony learning. *Pattern Recognition*, 42(11):2659–2670, 2009. doi: 10.1016/j.patcog.2009.03.029.

K. D. Majeske. A mixture model for automobile warranty data. *Reliability Engineering & System Safety*, 81(1):71–77, 2003. doi: 10.1016/S0951-8320(03)00073-5.

G. McLachlan and S. X. Lee. Emmixuskew: An r package for fitting mixtures of multivariate skew t distributions via the em algorithm. *Journal of Statistical Software*, 55(12):1–22, 2013. URL `http://www.jstatsoft.org/v55/i12`.

G. McLachlan and D. Peel. Contribution to the discussion of paper by s. richardson and p.j. green. *Journal of the Royal Statistical Society B*, 59(4):779–780, 1997. URL `http://www.jstor.org/stable/2985194`.

G. McLachlan and D. Peel. The emmix algorithm for the fitting of normal and t-components. *Journal of Statistical Software*, 4(2):1–14, 1999. URL `http://www.jstatsoft.org/v04/i02`.

G. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, New York, 2000.

V. Melnykov, W.-C. Chen, and R. Maitra. Mixsim: An r package for simulating data to study performance of clustering algorithms. *Journal of Statistical Software*, 51(12):1–25, 2012. URL `http://www.jstatsoft.org/v51/i12`.

M. Nagode. *rebmix: The Rebmix Package*, 2014. URL `http://CRAN.R-project.org/package=rebmix`. R package version 2.6.2.

M. Nagode and M. Fajdiga. A general multi-modal probability density function suitable for the rainflow ranges of stationary random processes. *International Journal of Fatigue*, 20(3):211–223, 1998. doi: 10.1016/S0142-1123(97)00106-0.

M. Nagode and M. Fajdiga. An improved algorithm for parameter estimation suitable for mixed weibull distributions. *International Journal of Fatigue*, 22(1):75–80, 2000. doi: 10.1016/S0142-1123(99)00112-7.

M. Nagode and M. Fajdiga. An alternative perspective on the mixture estimation problem. *Reliability Engineering & System Safety*, 91(4):388–397, 2006. doi: 10.1016/j.ress.2005.02.005.

M. Nagode and M. Fajdiga. The rebmix algorithm for the univariate finite mixture estimation. *Communications in Statistics - Theory and Methods*, 40(5):876–892, 2011a. doi: 10.1080/03610920903480890.

M. Nagode and M. Fajdiga. The rebmix algorithm for the multivariate finite mixture estimation. *Communications in Statistics - Theory and Methods*, 40(11):2022–2034, 2011b. doi: 10.1080/03610921003725788.

M. Nagode, J. Klemenc, and M. Fajdiga. Parametric modelling and scatter prediction of rainflow matrices. *International Journal of Fatigue*, 23(6):525–532, 2001. doi: 10.1016/S0142-1123(01)00007-X.

C. K. Reddya and B. Rajaratnam. Learning mixture models via component-wise parameter smoothing. *Computational Statistics and Data Analysis*, 54(3):732–749, 2010. doi: 10.1016/j.csda.2009.04.012.

C. Sharpsteen and C. Bracken. *tikzDevice: A Device for R Graphics Output in PGF/TikZ Format*, 2013. URL `http://CRAN.R-project.org/package=tikzDevice`. R package version 0.7.0.

K. S. Sultan, M. A. Ismail, and A. S. Al-Moisheer. Mixture of two inverse weibull distributions: Properties and estimation. *Computational Statistics & Data Analysis*, 51(11):5377–5387, 2007. doi: 10.1016/j.csda.2006.09.016.

A. E. Touw. Bayesian estimation of mixed weibull distributions. *Reliability Engineering & System Safety*, 94(2):463–473, 2009. doi: 10.1016/j.ress.2008.05.004.

M. Wiper, D. R. Insua, and F. Ruggeri. Mixtures of gamma distributions with applications. *Journal of Computational and Graphical Statistics*, 10(3):440–454, 2001. URL `http://www.jstor.org/stable/1391098`.

Marko Nagode
University of Ljubljana
Faculty of Mechanical Engineering
Aškerčeva 6
1000 Ljubljana
Slovenia
Marko.Nagode@fs.uni-lj.si.